

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

Калин Сергей Александрович

Выпускная квалификационная работа бакалавра

Математические модели влияния в социальных сетях

Направление 010400

Прикладная математика и информатика

Научный руководитель,

доцент

Лежнина Е.А.

Санкт-Петербург

2017

Содержание

Введение	3
Постановка задачи	5
Обзор литературы.....	6
Глава 1. Математическая модель.....	10
Глава 2. Определение наиболее влиятельных агентов в сети.....	16
2.1 Ценность агентов.....	16
2.2 Различия между влиятельностью и ценностью агентов	20
2.3 Поиск наиболее ценных агентов	21
2.4 Алгоритм поиска наиболее влиятельных агентов без определения ценности	23
Пример	26
Выводы	30
Заключение	31
Список литературы	32
Приложения	34

Введение

Социальные сети плотно укрепились в жизни людей. Сегодня трудно представить себе человека, не зарегистрированного в социальных сетях. Но социальные сети – это не только те сайты, которые мы регулярно посещаем в интернете. Любая группа людей, взаимодействующих между собой, образует социальную сеть. В науке эта тенденция привела к появлению нового направления для исследований.

В данной работе представлен обзор моделей влияния в социальных сетях. Кроме того, разработана новая математическая модель компьютерной социальной сети и способы нахождения самого влиятельного или группы наиболее влиятельных агентов. Компьютерные социальные сети в настоящее время играют большую роль в жизни современного человека. Практически каждый имеет аккаунт в компьютерных социальных сетях. Они являются одним из каналов продвижения продуктов, поиска новых талантов, заработка, общения, распространения информации и даже формирования общественного мнения. Поэтому исследование компьютерных социальных сетей на сегодняшний день является необходимой и актуальной задачей.

Под социальной сетью, в том числе и компьютерной, будем понимать структуру, которая состоит из множества агентов (пользователей) и определенного на ней множества отношений (влияний между агентами). Формально социальную сеть можно представить как граф $G(N, E)$, в котором $N = \{1, \dots, n\}$ – множество вершин (агентов), а E – множество ребер (влияний), которые отражают взаимодействие агентов.

Особое внимание нужно уделить влиянию в компьютерных социальных сетях. Под влиянием будем понимать процесс воздействия одним агентом (пользователем) на мнение другого агента и, как результат, – изменение мнения последнего. На сегодняшний день составлено множество моделей влияния в социальных сетях. Некоторые из них [1], [2] использовались для создания модели, представленной в данной работе.

Взаимодействуя в компьютерных социальных сетях, одни агенты

влиять на мнения других, тем самым побуждая последних к определенным действиям. Так, если найти группу самых влиятельных агентов, то благодаря им можно увеличить продажи компании в несколько раз, сформировать определенное мнение большинства или просто распространить нужную информацию за короткое время и при минимальных затратах. В настоящей работе построена модель, максимально приближенная к реальности, поскольку в последнее время эта проблема стала актуальной. Эта модель послужит основой для дальнейших исследований.

Постановка задачи

Цели работы. В рамках данной работы поставлены следующие цели:

- Проанализировать разные классы моделей влияния в социальных сетях;
- Разработать модель влияния в социальной сети. Модель должна быть максимально приближена к реальности;
- Найти алгоритмы нахождения самого влиятельного или группы влиятельных агентов сети;
- Написать программную реализацию алгоритмов;
- Реализовать алгоритмы на примере.

Обзор литературы

На сегодняшний день множество работ посвящено исследованию моделей социальных сетей, в том числе и моделям влияния. В одних из них больше внимания уделяется самой сети влияний, в других процессам и правилам взаимодействия. Анализ статей позволил представить в работе краткий обзор основных моделей социальных сетей, которые можно разделить на следующие группы.

1.1. Модели с порогами. Главной особенностью данного класса моделей является наличие порогов $\varphi_j \in [0,1]$, которые могут быть как линейными [3], так и нелинейными [4]. Агент в этой модели представлен как узел социальной сети. Он может изначально находиться только в одном из двух состояний: активном или неактивном. Причем обычно предполагается переход только из неактивного состояния в активное. Агенты взаимодействуют и влияют друг на друга с определенными степенями влияния w_{ij} . Если активный агент i влияет на неактивного агента j с величиной влияния w_{ij} , то агент j активируется при следующем условии активации: $\sum_i w_{ij} > \varphi_j$.

Величина порога зависит от самой математической модели: в одном случае значения порога могут фиксироваться для всех агентов [5], в другом определяться случайно согласно некоторому вероятностному закону распределения [6], в общем же индивидуальные различия основываются на личных качествах агента, его заинтересованности и т.п. [7].

1.2. Модели независимых каскадов. В этой группе моделей агенты сети также могут находиться только в одном из двух состояний. При активации агента i в момент времени t , с некоторой вероятностью p_{ij} на следующем шаге у него появляется возможность активировать каждого из своих соседей j . На следующем шаге уже активированные соседи с некоторой вероятностью могут активировать своих соседей и т. д.

Обобщение моделей 1.1 и 1.2. представлено работе [5].

1.3. Модели просачивания и заражения. Данная группа моделей является

одним из популярных способов изучения распространения информации в социальной сети.

Классическая модель строится на цикле заболевания носителя (агента). Изначально агент восприимчив к заболеванию. При вхождении в контакт с инфицируемым агентом рассматриваемый агент может быть заражен с некоторой вероятностью. Через некоторое время агент выздоравливает, приобретая при этом иммунитет к вирусу или умирает. Иммунитет со временем уменьшается, и агент опять становится восприимчивым.

В социальных сетях, например, это может происходить следующим образом. Человек написал блог о какой-нибудь проблеме. Другой прочитал этот блог (восприимчив). Решил написать об этой проблеме (заражен). Потом ему стало это неинтересно (иммунитет). Но через какое-то время опять вернулся к этой проблеме (восприимчив).

Одним из главных показателей моделей просачивания и заражения является критическая вероятность инфицирования соседнего агента – «эпидемический порог» β , если порог превышен, то заражение распространится на всю сеть. Распространение заражения зависит от модели социальной сети. Эпидемический порог может отсутствовать, например, как в сетях без масштаба. Если в них появляется инфекция, то она распространится на всю сеть [8].

1.4. Модели Изинга. Модель Изинга – математическая модель, учитывающая только взаимодействие ближайших атомов. С её помощью также могут моделироваться и социальные сети. Подобная задача была рассмотрена в статье [9]. Её авторы предполагают, что благодаря модели Изинга можно смоделировать независимость в большой группе. Определяющим является взаимодействие между ближайшими соседями в сети. Влияние авторитета - внешнее поле для социальной группы. А способность группы мыслить, принимать новые идеи служит аналогом температуры.

1.5. Модели на основе клеточных автоматов. Эта группа моделей также

является одним из популярных способов представления распространения информации в сети. Социальную сеть представляют, как сложную систему, которая состоит из множества агентов, взаимодействующих между собой. Это взаимодействие влияет на коллективное поведение агентов, анализ или предсказание которого вызывает большие сложности.

Набор агентов клеточного автомата составляет регулярную решетку. Текущее состояние агента характеризуется переменной, которая определяется в каждый момент времени. Состояния объекта меняются согласно определенным вероятностным правилам, зависящим от ближайших соседей и, возможно, даже от выбранного агента, через конечные интервалы времени.

Пример использования моделей показан в статье [10]. В этой работе представлен эффект «из уст в уста» в распространении информации в социальной сети. Агент соединяется сильными связями с агентами, входящими в его собственную сеть. Помимо этого, у агента имеются ещё и слабые связи. С помощью их он связан с агентами других собственных сетей. Несмотря на то, что вероятность того, что информация распространится по слабым связям ниже, чем по сильным, авторы приходят к следующему выводу: на скорость распространения информации слабые связи влияют, по крайней мере, в такой же степени, как и сильные.

1.6. Модели на основе цепей Маркова. Пример использования цепей Маркова при разработке моделей социальных сетей подробно рассмотрен в статье [11]. В работе описывается модель, в которой рассматриваются взаимодействия между агентами. Причем моделируются не только действия каждого агента, но и действия всей группы в целом. Такая структура называется двухуровневой.

В любой момент времени t каждый агент находится в определенном состоянии S_t^i . Вероятность этого состояния зависит как от предыдущего состояния агента, то есть его состояния в предыдущий момент времени, так и от состояния всей команды. Агент предпринимает действие O_t^i с вероятностью $P(O_t^i | S_t^i)$. В свою очередь команда также в некоторый момент времени t

находится в некотором состоянии S_t^G . Вероятность этого состояния зависит от состояний всех агентов. Авторы работы предполагают, что многоуровневое влияние может являться средством анализа социальной динамики для выявления шаблонов возникающего группового поведения.

В группах моделей 1.3-1.6 рассматриваются правила взаимодействия агентов. В тоже время мало внимания уделяется самой сети влияния, её свойствам, структурам и процессам взаимодействия, что является одним из недостатков этих моделей.

Моделирование влияния в социальных сетях коснулось и теории игр. Существует ряд теоретико-игровых моделей сетей:

- 2.1 Модели взаимной информированности [12]
- 2.2 Модели согласованных коллективных действий [13]
- 2.3 Модели коммуникаций [14]
- 2.4 Модели стабильности сети [15]
- 2.5 Модели информационного влияния и управления [16]
- 2.6 Модели информационного противоборства [17]

В этих моделях упор делается на взаимосвязь между агентами и их информированность. Агент поступает так, чтобы максимизировать свой выигрыш, но его выгода также зависит и от действий других агентов.

Глава 1. Математическая модель

Для начала составим математическую модель социальной сети. За её основу возьмем модели, предложенные в работах [1], [2] в которых рассматривается влияние агентов в социальных сетях на формирование мнений друг друга.

Пусть в нашу сеть вводится нововведение, к которому агенты могут относиться по-разному: положительно, нейтрально или отрицательно.

Пусть $N = \{1...n\}$ – множество агентов, которые входят в компьютерную социальную сеть. Любой человек всегда имеет некоторое собственное мнение по поводу нового товара, услуги или нововведения. Поэтому в модели каждый агент в начальный момент времени τ_0 имеет некоторое мнение относительно нововведения $b_i^{\tau_0}$. Начальное мнение всех агентов в сети мы обозначим как вектор-столбец неотрицательных начальных мнений b^0 размерности n . Будем считать, что $b_i^{\tau_0} \in [0; 1]$.

В каждый дискретный момент времени мнения агентов могут меняться. Так мнение агента в текущий момент времени может отличаться от мнения того же агента в предыдущий момент времени. Мнение i -го агента в момент времени τ будем обозначать b_i^τ , где $b_i^\tau \in [0; 1]$. Вектор мнений агентов в момент времени τ будем обозначать b^τ .

Обозначим пороги $h_{1i} \in [0; 1]$, $h_{2i} \in [0; 1]$, где h_{1i} – будет обозначать нижний порог. Если мнение агента опускается ниже этого порога, то агент начинает относиться к нововведению отрицательно. h_{2i} – будет обозначать верхний порог. Если мнение агента поднимается выше этого порога, то агент начинает относиться к нововведению положительно. Если же величина мнения агента находится между верхним и нижним порогами, то его отношение к нововведению будет нейтральным. Таким образом, если в какой-то момент времени τ выполняется следующее неравенство: $b_i^\tau \geq h_{2i}$, то будем считать, что агент i в данный момент времени положительно относится к нововведению, если $b_i^\tau \leq h_{1i}$, то будем считать, что агент в данный момент

отрицательно относится к нововведению. Если же $h_{1i} < b_i^T < h_{2i}$, то будем считать, что агент в данный момент t относится к нововведению нейтрально. Все люди уникальны, поэтому каждый агент социальной сети имеет свои индивидуальные нижний и верхний пороги, которые зависят от его личных качеств, доверия и побуждений.

Агенты внутри сети влияют друг на друга, причем i -ый агент влияет на j -го с одной силой, на k -го с другой и т. д. Степени влияния агентов будем задавать матрицей A размерности $n \times n$ с компонентами a_{ij} , где $a_{ij} \geq 0$ будут обозначать силу влияния i -го агента на j -го агента (или степень доверия j -го агента i -му).

Возможен случай, когда агенты не связаны друг с другом, то есть не влияют друг на друга. Тогда компоненты матрицы, соответствующие влиянию этих агентов друг на друга, будут равны: $a_{ij} = 0$.

Величина влияния агентов зависит от множества факторов. Этими факторами могут быть дружеские отношения, общие интересы, когда один агент является кумиром другого и т.д. При составлении модели будем использовать самые основные факторы, с помощью которых можно определить степень влияния агентов.

Введём факторы, определяющие влияние одних агентов на других: $x_{lij} \in [0; 1]$, $l = \{1...8\}$.

Здесь x_{1ij} – количество сообщений между i -ым и j -ым агентами: чем больше люди общаются друг с другом, тем большее влияние они друг на друга оказывают.

x_{2ij} – множество общих диалогов i -го и j -го агентов, также, как и в случае с количеством сообщений между i -ым и j -ым агентами, показывает, как часто люди общаются друг с другом. Соответственно, чем выше показатель, тем большее влияние i -ый агент оказывает на j -ый.

x_{3ij} – общие группы i -го и j -го агентов. Большое количество общих групп говорит об общих интересах агентов. Этот показатель в отдельности не

является самым ключевым, но при одновременно повышенных показателях предыдущих факторов имеет большое значение при определении степени влияния между агентами.

x_{4ij} – общие аудиозаписи i -го и j -го агентов. Также, как и общие группы, говорит об общих интересах и в отдельности не играет большой роли, но в совокупности с первыми двумя факторами имеет в некоторых случаях довольно большой вес.

x_{5ij} – общие друзья у i -го и j -го агентов. В отдельности этот показатель показывает общий круг друзей агентов и степень информированности друг о друге. При повышенных показателях предыдущих факторов говорит о высокой вероятности тесных отношений между агентами и, следовательно, о повышенном влиянии i -го агента.

x_{6ij} – статус i -го агента для j -го (лучшие друзья, родственные связи) (если указаны). Этот фактор может напрямую показывать степень доверительных отношений между агентами.

x_{7ij} – количество подписчиков у i -го агента. При большом количестве подписчиков велика вероятность, что агент является известным человеком и соответственно оказывает повышенное влияние на своих кумиров.

$x_{8ij} \in [\{0\}; \{1\}]$ – использование i -ым агентом нововведения (1 – использовал, 0 – не использовал). Обычно ко мнению человека, который уже пользуется нововведением, люди прислушиваются больше, чем к человеку, который об этом нововведении только слышал, поэтому этот фактор имеет большое значение.

Каждый фактор имеет свой вес при определении степени влияния между агентами. Причем этот вес в разных случаях может быть определен по-разному. Поэтому для правильного определения степени влияния между агентами необходимо ввести вектор коэффициентов, который бы помог в правильных пропорциях установить зависимость степеней влияния между агентами от выбранных факторов. Введём этот вектор коэффициентов α с

компонентами $\alpha_l \in [0; 1]$, $l = \{1...8\}$. Каждая компонента вектора коэффициентов соответствует компоненте вектора факторов, определяющих влияние. При этом компоненты вектора коэффициентов являются субъективными величинами, которые меняются в зависимости от нововведений. Так, например, при нововведении, касающегося определенного стиля музыки, коэффициент α_4 , соответствующий фактору общих аудиозаписей, будет выше, чем такой же коэффициент при нововведении, связанным с технологическими новинками.

Теперь, когда определены основные факторы и их веса в определении степени влияний, можно определить влияния агентов друг на друга. Итак, степень влияния i -го агента на j -го будет прямо пропорциональна компонентам вектора факторов и компонентам вектора коэффициентов, соответствующим этим факторам, и вычисляться по формуле:

$$a_{ij} = \sum_{l=1}^8 \alpha_l x_{lij}$$

Таким образом, так как на j -го агента воздействуют n остальных агентов (их влияние может равняться 0), то общее влияние всех агентов, которые напрямую связаны с агентом j , на этого агента j будет вычисляться по формуле:

$$a_{iJ} = \sum_{j=1}^n \sum_{l=1}^8 \alpha_l x_{lij}$$

Можно заметить, что мы столкнулись с некоторой неточностью. Так как α_l и x_{lij} меньше или равны единице, то и a_{ij} не будет превосходить единицу. Тогда, ввиду того, что степень влияния агентов не превышает единицу, возникает сложность правильного определения мнений агентов посредством влияний в дискретные моменты времени τ .

Чтобы избежать данной проблемы, определим окончательное влияние i -го агента на j -го. Для этого нам понадобятся верхний и нижний пороги $h_{1i} \in [0; 1]$ и $h_{2i} \in [0; 1]$. Если мнение i -го агента ниже нижнего порога $h_{1i} \in [0; 1]$,

то этот агент отрицательно относится к нововведению. Тогда его влияние должно уменьшать мнение j -го агента, для этого достаточно задать эту степень влияния следующей формулой: $1 - a_{ij}$. Если собственное мнение i -го агента лежит между нижним и верхним порогами ($h_{1i} < b_i < h_{2i}$), то этот агент относится к нововведению нейтрально. Так как агент нейтрально относится к нововведению, то его влияние на агента j не должно менять собственного мнения последнего (b_j^τ) в этом случае степень влияния будет равняться 1. Наконец, если мнение i -го агента выше верхнего порога $h_{2i} \in [0; 1]$, то этот агент положительно относится к нововведению. Значит его влияние должно увеличивать собственное мнение агента j относительно нововведения и в этом случае степень влияния будет равняться $a_{ij} + 1$. Таким образом, у нас получается новая матрица окончательных влияний размерности $n \times n$ с компонентами a_{ij}^+ , a_{ij}^- и a_{ij}^0 , которые будут вычисляться по формулам:

$$a_{ij}^+ = a_{ij} + 1, \text{ если агент } i \text{ положительно относится к нововведению}$$

$$a_{ij}^- = 1 - a_{ij}, \text{ если агент } i \text{ отрицательно относится к нововведению}$$

$$a_{ij}^0 = 1, \text{ если агент } i \text{ нейтрально относится к нововведению}$$

При этом будет выполняться свойство транзитивности: если агент с номером j влияет на агента с номером i , и агент i влияет на агента с номером k , то это означает, что агент j косвенно влияет на агента k и т.д. Таким образом, возможно образование цепочки косвенных влияний.

В сети агенты взаимодействуют друг с другом, то есть происходит обмен мнениями. Это взаимодействие приводит к изменению мнения одних агентов под влиянием других.

Величина мнения j -го агента в момент времени τ равна:

$$b_j^\tau = b_j^{\tau-1} + \left(\sum_{i \in N^+} (a_{ij}^+ b_j^{\tau-1} - b_j^{\tau-1}) - \sum_{i \in N^-} (b_j^{\tau-1} - a_{ij}^- b_j^{\tau-1}) \right)$$

Если окажется, что $b_j^\tau < 0$, то будем считать, что мнение этого j -го агента в момент времени τ равно 0 ($b_j^\tau = 0$).

При этом если $b_j^\tau \geq h_{2j}$ то агент с номером j будет относиться к нововведению в момент времени τ положительно. Если $b_j^\tau \leq h_{1j}$ то агент j будет относиться к нововведению отрицательно. Если $h_{1j} < b_j^\tau < h_{2j}$, то агент j будет относиться к нововведению нейтрально.

Тогда, как показано в [2], в конечном итоге мнения агентов сходятся к результирующему вектору мнений $B = \lim_{\tau \rightarrow \infty} b^\tau$. Можно записать соотношение:

$$B = b + ((A^+b - b) - (b - A^-b))$$

$$\text{где } A^- = \lim_{\tau \rightarrow \infty} (a^-)^\tau, A^+ = \lim_{\tau \rightarrow \infty} (a^+)^\tau.$$

Глава 2. Определение наиболее влиятельных агентов социальной сети

Мы описали модель влияния в компьютерной социальной сети. Теперь нам необходимо найти наиболее влиятельного агента или группу агентов. При этом следует учитывать, что часто необходимость влияния связана с ограничением по времени. Например, выпускается новая коллекция зимней одежды. Продавцы хотят сделать рекламную кампанию и выбирают агента для предоставления ему скидки. При этом им необходимо, чтобы максимальное число покупок было сделано до конца февраля, потому что дальше одежда будет не востребована.

2.1 Ценность агентов

Найти наиболее влиятельного агента сети можно несколькими способами. Один из них требует определения ценности агента.

Рассмотрим социальную сеть, как потенциальных потребителей некоторого товара, услуги или потенциальных последователей некоторого нововведения. Определим ценность этих потребителей. С точки зрения продавца ценность агента в этой социальной сети будет определяться согласно следующим пунктам:

1. Ожидаемой прибылью от продажи товара, услуги или нововведения непосредственно самому агенту.
2. Влиянием агента на других агентов социальной сети и изменения их мнения о товаре, услуги или нововведении. Ожидаемой прибылью от агентов социальной сети, на которых влияет данный агент.

Зачастую в социальной сети ищут не одного самого влиятельного агента, а группу наиболее влиятельных агентов, для того чтобы предоставить им скидки, льготы или специальные акции, способствующие наибольшему распространению товара, услуги или нововведения по всей социальной сети.

При определении ценности агентов мы будем использовать результаты работы [16]. Авторы этой работы обращаются к проблеме нахождения k – наиболее влиятельных агентов. Ценность агента определяется не только ожидаемой прибылью от продажи продукта непосредственно этому агенту, но и ожидаемой прибылью от продажи продукта и тем агентам, на которых он оказывает влияние, от ожидаемой прибыли от продаж агентам, на которых они могут повлиять и т. д.

Также, как и в работе [16] поставим задачу нахождения оптимальной маркетинговой кампании $MA = \{MA_1, MA_2, \dots, MA_n\}$, где MA_i – это маркетинговая кампания для i -го агента и может принимать значения: 1 – наличие маркетингового воздействия, 0 – отсутствие маркетингового воздействия. MA_i определяется для всех n агентов социальной сети с предикатом $X_i = 1$ если i -ый агент купил товар, услугу или использует нововведение и $X_i = 0$ в противном случае. Будем считать, что значение X_i известно для всех агентов в нашей социальной сети. Предположим, что товар, услуга или нововведение имеет множество атрибутов $Y = \{Y_1, Y_2, \dots, Y_m\}$, где m количество атрибутов. Используя результаты работы [16] определим ценность агента нашей социальной сети.

Пусть c_i – затраты на маркетинговую кампанию для одного агента i социальной сети. От агентов на которых действует маркетинговая кампания и от агентов, на которых она не действует выручка может быть разной. Выручку от продажи товара, услуги или нововведения агенту, для которого была проведена маркетинговая кампания, будем обозначать rv_1 . Выручку от продажи товара, услуги или нововведения агенту, для которого не была проведена маркетинговая кампания, будем обозначать rv_0 . Если маркетинговая кампания будет включать скидку для агента, то справедливо $rv_1 < rv_0$.

Будем считать $f_i^1(MA)$ – величина, соответствующая проведению маркетинговой кампании для агента i , т.е. $MA_i=1$. $f_i^0(MA)$ – величина,

соответствующая отсутствию маркетинговой кампании для агента i .

Вычислим ожидаемую прибыль при воздействии рекламной кампании для одного агента. Ожидаемая прибыль от проведения маркетинговой кампании для агента i без учета её воздействия на других агентов и без учета влияния агента на других агентов будет определяться следующей формулой:

$$ELP_i(B, Y, MA) = rv_i P(X_i = 1 | b_i, Y, f_i^1(MA)) - rv_0 P(X_i = 1 | b_i, Y, f_i^0(MA)) - c_i,$$

где $P(X_i = 1 | b_i, Y, f_i^1(MA))$ – условная вероятность покупки i -ым агентом товара, услуги или нововведения при проведении маркетинговой кампании для этого агента, а $P(X_i = 1 | b_i, Y, f_i^0(MA))$ – условная вероятность покупки i -ым агентом товара, услуги или нововведения при отсутствии проведения для него маркетинговой кампании.

Очевидно, что при увеличении значения собственного мнения агента вероятность того, что он купит товар, услугу или станет использовать нововведение увеличивается. Будем считать, что наличие маркетинговой кампании повышает мнение агента и, следовательно, может только увеличить вероятность покупки товара, услуги или использования нововведения.

Если нам надо оценить ценность не одного, а группы агентов, то ожидаемая прибыль от проведения маркетинговой кампании для k агентов социальной сети без учета её воздействия на других агентов и без учета влияния этих k агентов на других агентов, будет находиться по формуле:

$$\sum_{i=1}^k ELP_i(B, Y, MA) = \sum_{i=1}^k rv_i P(X_i = 1 | b_i, Y, f_i^1(MA)) - \sum_{i=1}^k rv_0 P(X_i = 1 | b_i, Y, f_i^0(MA_0)) - \sum_{i=1}^k c_i,$$

где MA_0 – нулевой вектор.

При воздействии на агента с помощью маркетинговой кампании его мнение о товаре, услуге или нововведении меняется. Если мнение одного агента в социальной сети меняется, то благодаря его влиянию на других агентов, меняются мнения о товаре, услуге или нововведении и у других агентов социальной сети. В свою очередь эти агенты влияют на других

агентов и т.д. Соответственно, если мы хотим найти ценность агента, то нам необходимо учитывать косвенное влияние этого агента на всех других агентов сети.

Распространяя предыдущие выводы на нашу социальную сеть и учитывая косвенное влияние агентов получаем ожидаемое повышение прибыли от маркетинговой кампании в нашей сети, при воздействии на некоторую группу агентов:

$$ELP(B, Y, MA) = \sum_{i=1}^n rv_i P(X_i = 1 | b_i, Y, MA) - \sum_{i=1}^n rv_0 P(X_i = 1 | b_i, Y, MA_0) - \sum_{i=1}^k c_i,$$

где n – количество агентов в сети, $rv_i = rv_1$, если $MA_i=1$ и $rv_i = rv_0$, если $MA_i=0$.

Очевидно, что эта формула распространяется и для проведения маркетинговой кампании всего лишь для одного агента. Если маркетинговая кампания будет воздействовать только на агента под номером i , то вектор MA будет состоять из единицы на позиции под номером i и из нулей на всех остальных позициях. Тогда формула ожидаемого повышения прибыли будет выглядеть следующим образом:

$$ELP(B, Y, MA) = \sum_{i=1}^n rv_i P(X_i = 1 | b_i, Y, MA) + \sum_{i=1}^n rv_0 P(X_i = 1 | b_i, Y, MA_0) - c_i,$$

где n – будет также обозначать количество агентов в сети, $rv_i = rv_1$, при $MA_i=1$ и $rv_i = rv_0$, при $MA_i=0$.

Определим вероятность покупки i -ым агентом некоторого нововведения. Как уже было сказано каждый агент имеет свое собственное мнение о новом продукте или услуге. При этом даже при огромном желании купить это нововведение у него может попросту не оказаться возможности это сделать. Например, недостаток денег, наличие уже такого же или аналогичного товара и т.д. Этот факт никак нельзя упускать при расчете ожидаемой прибыли. Поэтому необходимо ввести вектор возможностей L , состоящий из нулей и единиц. 0 на i -ом месте говорит о том, что агент i не имеет возможность купить нововведение, 1- имеет.

В тоже время нам пока неизвестно, как именно влияет на вероятность покупки маркетинговая кампания, будем считать следующее:

- если мнение агента $b_i^\tau \leq h_{1i}$, то при воздействии на него маркетинговой кампании его мнение увеличивается на h_{2i} , а величина влияния увеличивается на $2a_{ij}$.
- если мнение агента $h_{1i} < b_i^\tau < h_{2i}$, то при воздействии на него его мнение увеличивается на величину $h_{2i} - b_i^\tau$, а величина влияния становится равной $a_{ij} + 1$.
- если мнение агента $h_{2i} \leq b_i^\tau$, то при воздействии на него его мнение увеличивается до 1, а влияние остается прежним.

Таким образом, вероятность покупки агентом нововведения будет вычисляться по формуле:

$$P(X_i = 1|b_i, Y, MA) = b_i^{MA\tau} \times l_i$$

где $b_i^{MA\tau}$ будет обозначать мнение агента о нововведении с учетом вектора маркетинговой кампании MA в момент времени τ , l_i – возможность покупки, то есть компоненту вектора L .

Будем считать агента i социальной сети наиболее ценным, если ожидаемое повышение прибыли от всех агентов сети при воздействии на агента i маркетинговой кампании будет являться наибольшим среди всех возможных ожидаемых повышений прибыли при воздействии маркетинговой кампании на агентов сети поодиночке.

Будем считать группу из k агентов социальной сети наиболее ценной, если ожидаемое повышение прибыли от всех агентов сети при воздействии на эту группу маркетинговой кампании будет являться наибольшим среди всех возможных ожидаемых повышений прибыли при воздействии маркетинговой кампании на всевозможные группы этой сети.

2.2 Различия между влиятельностью и ценностью агентов

Мы определили ценность агентов сети. Она определяется с помощью полученной информации об ожидаемой прибыли данного агента, и агентов на

которых распространяется его влияние.

Изначально все находятся в равных условиях. То есть, маркетинговая кампания не проводится ни для одного агента сети. При определении ожидаемого повышения прибыли предполагается, что для какого-то агента сети производится маркетинговое воздействие. Затраты на маркетинговую кампанию при определении ценности агентов одинаковы для каждого из них. Соответственно, стоимость (rv_1) товара, услуги или нововведения для всех агентов, находящихся под влиянием маркетинговой кампании одинакова. Стоимость (rv_0) товара для тех, кто не попал под данное воздействие, может отличаться от стоимости rv_1 , но она так же одинакова для всех агентов образовавшейся группы.

В тоже время, маркетинговая деятельность оказывает различное влияние на агентов в зависимости от значения их мнения к нововведению. Если мнение отрицательное, то оно повышается на величину верхнего порога агента, а влияние увеличивается на $2a_{ij}$. Если нейтральное то увеличение происходит до верхнего порога, а влияние на величину a_{ij} . Если агент положительно относится к нововведению, то его мнение повысится до 1, а величина влияния не изменится.

Таким образом, для компании большую ценность могут представлять агенты с отрицательным мнением о продукте, а агенты с положительным мнением могут быть не так ценны, поскольку они и так будут оказывать положительное воздействие на всю сеть.

2.3 Поиск наиболее ценных агентов

Теперь, когда мы определили различия между ценностью агента и его влиятельностью, можно перейти к поиску наиболее ценного агента или группы агентов сети. Их нахождение будет строиться по следующему алгоритму.

Первый шаг. Изначально агенты ничего не знают о товаре, услуге или нововведении. Поэтому до появления нового товара, услуги или нововведения величина их мнения об этих продуктах равна нулю. На первом шаге агенты

узнают о новом товаре, услуге, или нововведении. После этого формируется мнение b_i каждого агента об этом товаре, услуге или нововведении. Формируется матрица влияний агентов с компонентами a_{ij} (так как влияние зависит от покупки продукта). Задается количество моментов времени τ .

Второй шаг. На втором шаге все агенты начинают влиять на мнения друг друга. Каждый агент влияет на соседа с силой, соответствующей компонентам матрицы влияний. Агенты взаимодействуют постоянно. Поэтому в каждый момент времени τ мнения агентов относительно товара, услуги или нововведения будут меняться.

Третий шаг. Когда мнение каждого агента сформировано, влияние агентов друг на друга определено, можно найти наиболее ценного агента или группу агентов социальной сети. Для нахождения этих агентов в сети нужно найти такое MA , которое максимизировало бы величину ожидаемой прибыли ELP . При этом если мы хотим найти какое-то конкретное число агентов, то количество компонент вектора MA равное единице, не должно превосходить k , где k - количество агентов, которых нам нужно найти. $k = 1$, если нам нужно найти одного наиболее ценного агента сети. $k = n$, где n - это количество агентов входящих в нашу группу, если нам нужно найти группу наиболее ценных агентов, состоящую из n человек. В общем случае это делается перебором всех возможных комбинаций. Возможны некоторые аппроксимационные подходы, которые дают приближенное решение.

- 1) Одиночный обход. Будем проводить маркетинговую кампанию для агента i , то есть $MA_i = 1$, если будет выполняться неравенство:

$$ELP(B, Y, MA) > 0$$

где вектор MA будет состоять из нулей за исключением позиции i , на которой будет стоять единица. В результате такого обхода выделяется k ценных агентов воздействие на которых приводит к повышению ожидаемой прибыли. Для сокращения числа агентов можно сравнить ожидаемое повышение прибыли и найти

наиболее ценного агента или группу агентов.

- 2) Жадный алгоритм. Пусть $MA = MA_0$. Будем обходить в цикле MA_i и устанавливать значение равное единице, то есть $MA_i = 1$, если

$$ELP(B, Y, f_i^1(MA)) > ELP(B, Y, MA)$$

Напомним, $f_i^1(MA)$ – это величина, соответствующая проведению маркетинговой кампании для агента i . Таким образом, мы найдем группу ценных агентов, которая приводит к повышению ожидаемой прибыли.

- 3) Поиск с восхождением по выпуклой поверхности. Пусть $MA = MA_0$. Будем устанавливать значение $MA_{i_1} = 1$, где $i_1 = \operatorname{argmax}_i(ELP(B, Y, f_i^1(MA)))$. Будем повторять эту операцию, пока существует агент i , установка которого на $MA_i = 1$ дает повышение ELP и пока количество найденных агентов меньше количества агентов, входящих в группу, которую нам надо найти (k). Таким образом, мы найдем нужное количество ценных агентов, к которым необходимо применить маркетинговое воздействие.

Первые два подхода находят группу ценных агентов сети. Причем один из них основывается на индивидуальном повышении ожидаемой прибыли, а другой на кооперативном. В отличие от них третий подход может обеспечивать заданную размерность этой группы. Таким образом, мы можем найти какое-то определенное количество ценных агентов.

2.4 Алгоритм поиска наиболее влиятельных агентов без определения ценности

Мы рассмотрели способ нахождения наиболее влиятельного агента или группы агентов социальной сети с точки зрения определения ценности. Теперь рассмотрим способ нахождения наиболее влиятельного агента или группы агентов сети без каких-либо дополнительных определений.

Мы знаем, что агенты каким-то образом влияют друг на друга. Причем

каждый агент может повлиять только на определенное количество других агентов, с кем у него установлена связь (с кем он контактирует). У каждого агента свое собственное число связей с другими агентами. Один может контактировать с пятью, другой с двадцатью, третий с пятнадцатью и т.д. Большое количество связей с другими агентами не гарантирует большую влияние агента. Например, агент i влияет на двадцать пять агентов, а агент j на десять. При этом двадцать пять агентов, на которых влияет агент i , больше ни на кого влияния не оказывают, а десять агентов на которых влияет агент j оказывают влияние ещё на пятьдесят двух агентов сети. Получается, что агент j может оказывать большее влияние, чем агент i .

Но это утверждение также остается под вопросом. Так как хоть агент j в итоге оказывает влияние на большее количество агентов социальной сети, чем агент i , но это влияние может быть слабее. То есть влияние агента j на мнения других агентов, с которыми он контактирует, может не иметь достаточного эффекта, и лишь малая часть агентов поменяют свое мнение. А влияние агента i на агентов, с которыми у него есть связь, наоборот поменяет мнения такого количества агентов, которое будет превосходить количество агентов, поменявших мнение под влиянием агента j .

Итак, мы должны учитывать, что большое количество связей и высокая сила влияния на малое число агентов, далеко не всегда являются признаками самого влиятельного агента в сети.

Рассмотрим нашу социальную сеть. У нас имеются n агентов и вектор их мнений относительно товара, услуги или нововведения. Для этих агентов составлена окончательная матрица влияний. Известно количество связей у каждого агента сети. Необходимо найти самого влиятельного агента.

Первый шаг. На первом шаге, так же, как и в алгоритме с определением ценности, агенты узнают о товаре, услуге или нововведении. После этого у них формируются собственные мнения b_i относительно этого продукта. Формируется матрица влияний агентов с компонентами a_{ij} .

Второй шаг. На этом шаге мы считаем количество связей между

агентами. И вычисляем величину влиятельности p для каждого агента.

Величина влиятельности i -го агента p_i это величина, равная сумме вероятностей агента i поменять мнение агентов, на которых влияет агент i , на мнение, совпадающее с агентом i , на мнение ближайшее к мнению агента i (например: мнение агента i положительное и под влиянием агента i агент j меняет свое отрицательное мнение на нейтральное) или укрепить мнение, совпадающее с агентом i , плюс сумма вероятностей агентов, на которых влияет агент i , поменять мнение агентов, на которых они влияют, на мнение, совпадающее с агентом i или на мнение ближайшее к мнению агента i и т.д.

$$p_i = \sum_{i=1}^m P(C = 1|B, A, MA) + \sum_{i=1}^z P(C = 1|B, A, MA) + \dots$$

где $C = 1$ обозначает смену мнения агента на мнение совпадающее с агентом i , укрепление мнения агента совпадающего с агентом i или смена мнения агента на мнение ближайшее к мнению агента i . $P(C = 1|B, A, MA)$ – вероятность смены мнения агента на мнение совпадающее с агентом i , укрепления мнения агента совпадающего с мнением агента i или смены мнения агента на мнение ближайшее к мнению агента i . m – количество агентов, на которые влияет агент i . z – количество агентов, на которые влияют агенты, на которых оказывает влияние агент i .

Таким образом, когда у нас рассчитаны величины влиятельности каждого агента социальной сети, мы можем путем перебора всех возможных комбинаций найти наиболее влиятельного агента или группу агентов социальной сети.

Для нахождения можно использовать аппроксимационные подходы 2 или 3, дающие приближенное решение, описывающиеся в пункте нахождения самого ценного агента или группы агентов.

Пример

Для описанных алгоритмов нахождения наиболее ценного агента или группы агентов составлена программа на языке программирования C#, представленная в приложении, а также реализован численный пример.

Программа рассчитана на любое количество агентов. Матрица влияний и вектор мнений меняются каждый дискретный момент времени. Количество этих моментов задается пользователем. Предлагается на выбор один из рассмотренных алгоритмов с определением ценности. Первые два находят группу ценных агентов социальной сети, маркетинговое воздействие на которую приведет к повышению ожидаемой прибыли. Третий алгоритм позволяет найти группу наиболее ценных агентов сети, размер которой определяет сам пользователь.

Реализован численный пример.

Рассмотрим 20 агентов. Каждый агент имеет свое собственное мнение о нововведении: 0,09; 0,19; 0,46; 0,04; 0,55; 0,73; 0,5; 0,61; 0,63; 0,67; 0,04; 0,18; 0,02; 0,19; 0,88; 0,98; 0,36; 0,37; 0,96; 0.

Для каждой пары агентов задано множество факторов, определяющих влияние $x_{lij} \in [0; 1]$, $l = 1 \dots 8$. Заданы коэффициенты $\alpha_l \in [0; 1]$, соответствующие этим факторам и составлена матрица начальных влияний (рис.1).

0	0,93	0,38	0,42	0,73	0,26	0,83	0,08	0,1	0,12	0,23	0,8	0,91	0,14	0,81	0,11	0,58	0,08	0,62	0,52
0,77	0	0,56	0,82	0,71	0,22	0,02	0,07	0,09	0,64	0,36	0,61	0,2	0,78	0,3	0,11	0,84	0,32	0,05	0,14
0,86	0,11	0	0,57	0,63	0,65	0,72	0,68	0,54	0,94	0,19	0,03	0,37	0,93	0,53	0,26	0,46	0,08	0,86	0,6
0,87	0,02	0,52	0	0,14	0,39	0,05	0,4	0,57	0,5	0,31	0,99	0,95	0,92	0,67	0,11	0,02	0,04	0,83	0,01
0,02	0,21	0,58	0,26	0	0,95	0,73	0,1	0,93	0,7	0,58	0,26	0,3	0,05	0,93	0,7	0,28	0,2	0,62	0,66
0,16	0,49	0,85	0,05	0	0,21	0,23	0,04	0,37	0,36	0,98	0,97	0,56	0,96	0,03	0,3	0,36	0,28	0,18	
0,15	0,85	0,53	0,86	0,59	0,31	0	0,52	0,59	0,35	0,03	0,38	0,97	0,53	0,66	0,21	0,53	0,27	0	0,83
0,13	0,87	0,24	0	0,06	0,05	0,25	0	0,54	0,56	0,96	0,77	0,82	0,58	0,78	0,6	0,36	0,04	0,56	0,74
0,46	0,66	0,84	0,17	0,87	0,04	0,25	0,31	0	0,83	0,77	0,23	0,62	0,83	0,63	0,38	0,19	0,15	0,21	0,06
0,88	0,83	0,46	0,11	0,67	0,11	0,2	0,26	0,18	0	0,05	0,3	0,63	0,71	0,51	0,19	0,98	0,89	0,73	0,57
0,13	0,36	0,4	0,03	0,34	0,26	0,34	0,76	0,24	0,72	0	0,41	0,16	0,56	0,99	0,35	0,83	0,32	0,28	0,45
0,56	0,2	0,17	0,56	0,15	0,37	0,23	0,45	0,77	0,86	0,81	0	0,42	0,02	0,39	0,91	0,64	0,37	0,44	0,1
0,78	0,41	0,84	0,93	0,11	0,44	0,9	0,37	0,16	0,43	0,21	0,52	0	0,62	0,96	0,88	0	0,96	0,26	0,53
0,02	0,79	0,97	0,42	0,41	0,68	0,44	0,92	0,07	0,83	0,24	0,16	0,79	0	0,59	0,01	0,16	0,52	0,83	0,99
0,04	0,97	0,74	0,45	0,56	0,53	0,87	0,49	0,2	0,48	0,12	0,51	0,59	0,8	0	0,91	0,43	0,02	0,31	0,18
0,05	0,79	0,77	0,35	0,65	0,84	0,78	0,59	0,49	0,58	0,32	0,31	0,22	0,55	0,55	0	0,65	0,14	0,54	0,84
0,02	0,44	0,03	0,31	0,6	0,7	0,73	0,7	0	0,37	0,21	0,52	0,18	0,22	0,51	0,43	0	0,83	0,5	0,56
0,44	0,5	0,84	0,06	0,94	0,36	0,65	0,21	0,88	0,71	0,09	0,99	0,04	0,51	0,21	0,34	0,45	0	0,58	0,26
0,8	0,37	0,41	0,98	0,77	0,85	0,18	0,17	0,81	0,56	0,75	0,95	0,33	0,2	0,65	0,02	0,85	0,24	0	0,8
0,15	0,21	0,38	0,45	0,79	0,17	0,8	0,58	0,02	0,53	0,37	0,71	0,32	0,81	0,99	0,22	0,28	0,2	0,91	0

(рис.1)

Для каждого агента задан верхний и нижний пороги, которые также играют роль в определении конечной матрицы влияний:

Номер агента	нижний	верхний	Номер агента	нижний	Верхний
1	0,3	0,7	11	0,3	0,8
2	0,4	0,8	12	0,3	0,8
3	0,2	0,7	13	0,25	0,8
4	0,25	0,8	14	0,25	0,7
5	0,3	0,7	15	0,4	0,8
6	0,4	0,8	16	0,2	0,6
7	0,2	0,7	17	0,3	0,7
8	0,3	0,6	18	0,3	0,7
9	0,3	0,7	19	0,3	0,8
10	0,4	0,8	20	0,3	0,7

Необходимо найти группу ценных агентов в сети, воздействие на которых приведет к повышению ожидаемой прибыли. А также 3-х наиболее ценных агентов для того, чтобы провести для них маркетинговую кампанию, в случае если окажется, что группа ценных агентов сети слишком велика. Количество моментов времени равно $\tau = 2$. Маркетинговая кампания включает в себя скидку, поэтому стоимость продукта с учетом маркетингового воздействия $rv_1 = 3$ у.е., без маркетингового воздействия $rv_0 = 5$ у.е., затраты на маркетинг для одного агента $c = 1$ у.е. Будем считать, что каждый элемент вектора возможностей агентов равен единице.

Решение:

Используя матрицу влияний и данные о порогах мнений агентов, составляем конечную матрицу влияний (рис.2).

0	0,07	0,62	0,58	0,27	0,74	0,17	0,92	0,9	0,88	0,77	0,2	0,09	0,86	0,19	0,89	0,42	0,92	0,38	0,48
0,23	0	0,44	0,18	0,29	0,78	0,98	0,93	0,91	0,36	0,64	0,39	0,8	0,22	0,7	0,89	0,16	0,68	0,95	0,86
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0,13	0,98	0,48	0	0,86	0,61	0,95	0,6	0,43	0,5	0,69	0,01	0,05	0,08	0,33	0,89	0,98	0,96	0,17	0,99
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1,13	1,87	1,24	1	1,06	1,05	1,25	0	1,54	1,56	1,96	1,77	1,82	1,58	1,78	1,6	1,36	1,04	1,56	1,74
1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0,87	0,64	0,6	0,97	0,66	0,74	0,66	0,24	0,76	0,28	0	0,59	0,84	0,44	0,01	0,65	0,17	0,68	0,72	0,55
0,44	0,8	0,83	0,44	0,85	0,63	0,77	0,55	0,23	0,14	0,19	0	0,58	0,98	0,61	0,09	0,36	0,63	0,56	0,9
0,22	0,59	0,16	0,07	0,89	0,56	0,1	0,63	0,84	0,57	0,79	0,48	0	0,38	0,04	0,12	1	0,04	0,74	0,47
0,98	0,21	0,03	0,58	0,59	0,32	0,56	0,08	0,93	0,17	0,76	0,84	0,21	0	0,41	0,99	0,84	0,48	0,17	0,01
1,04	1,97	1,74	1,45	1,56	1,53	1,87	1,49	1,2	1,48	1,12	1,51	1,59	1,8	0	1,91	1,43	1,02	1,31	1,18
1,05	1,79	1,77	1,35	1,65	1,84	1,78	1,59	1,49	1,58	1,32	1,31	1,22	1,55	1,55	0	1,65	1,14	1,54	1,84
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1,8	1,37	1,41	1,98	1,77	1,85	1,18	1,17	1,81	1,56	1,75	1,95	1,33	1,2	1,65	1,02	1,85	1,24	0	1,8
0,85	0,79	0,62	0,55	0,21	0,83	0,2	0,42	0,98	0,47	0,63	0,29	0,68	0,19	0,01	0,78	0,72	0,8	0,09	0

(рис.2)

Для начала найдем группу ценных агентов сети. Для этого реализуем первые два алгоритма, а именно алгоритмы, включающие в себя одиночный обход и жадный алгоритм. После реализации сравним результат.

Алгоритм с одиночным обходом.

При решении примера алгоритмом с одиночным обходом были отобраны номера агентов маркетинговое воздействие на которых повышает ожидаемую прибыль. Номера этих агентов: 13; 14; 17; 18; 20.

В результате маркетингового воздействия на отобранных агентов ожидаемая прибыль компании составит: 91,9191728 у.е.

Алгоритм включающий жадный алгоритм.

В ходе решения примера подходом, включающим жадный алгоритм, выявлены номера агентов совокупное воздействие на которых приведет к повышению ожидаемой прибыли. Номера этих агентов: 13; 14; 17; 18; 20.

При маркетинговой кампании включающей номера этих агентов ожидаемая прибыль также составит: 91,9191728 у.е.

Результаты совпадают. Оба алгоритма выбрали в качестве агентов, на которых необходимо произвести маркетинговое воздействие номера: 13; 14; 17; 18; 20. Но этого могло и не произойти, так как алгоритмы работают по разным принципам. Поэтому для более точного выявления группы ценных агентов необходимо проверять оба алгоритма впоследствии сравнивая

результаты.

Решим вторую часть задачи. Для этого воспользуемся третьим алгоритмом, включающим в себя поиск с восхождением.

Введем размерность группы 3 и рассмотрим результат.

Алгоритм показал, что тройка самых ценных агентов в сети: 20; 13; 4. И ожидаемая прибыль компании составит: 96.1778 у.е.

Включение в группу агента под номером 4 связано с максимизацией ожидаемой прибыли на каждом шаге, тогда как в предыдущие алгоритмы обеспечивали лишь её увеличение.

Вывод. Таким образом, агенты под номерами 13; 14; 17; 18; 20 образуют группу ценных агентов сети, маркетинговое воздействие на которую приведет к повышению ожидаемой прибыли. А самая ценная группа из 3 агентов включает в себя номера: 20; 13; 4.

Выводы

Основные результаты, полученные в процессе написания данной работы:

- ✓ Составлена новая математическая модель влияния в компьютерной социальной сети, приближенная к реальной жизни;
- ✓ Разработаны алгоритмы, позволяющие найти оптимальную группу агентов для проведения маркетинговых кампаний;
- ✓ Составлена программа на языке программирования C#;
- ✓ Реализован пример для 20 агентов.

Заключение

Роль социальных сетей в жизни людей сложно переоценить. Люди различных возрастных категорий общаются через социальные сети, такие как Вконтакте, Твиттер и др. При этом практически любой из них даже не осознает, что каждый день подвергается влиянию.

Разработка подробной модели влияния в социальных сетях, приближенной к реальности, помогает понять, как именно распространяется влияние в сети и на каких агентов нужно воздействовать, чтобы оно охватило наибольшую аудиторию.

Социальные сети давно стали инструментом распространения информации. Правильно воздействуя через них, люди формируют определенные общественные мнения и побуждают других на определенные действия.

Нельзя недооценивать возможность влияния различных компаний на агентов социальных сетей. При грамотном моделировании социальных сетей представляется возможность оптимизировать затраты на продвижение продуктов, повысить приток новых клиентов и, как следствие, в несколько раз увеличить уровень прибыли. Для этого необходимо лишь грамотно выбрать правильных агентов в сети, которые при минимальных расходах смогут повлиять на широкую аудиторию.

К сожалению, на сегодняшний день разработано не очень большое количество моделей социальных сетей, приближенных к реальной жизни. Также ощущается и недостаток алгоритмов нахождения влиятельных агентов, что является большим пространством для новых исследований.

Список литературы:

1. Калинин С.А. Модель влияния в компьютерной социальной сети // Научные труды sworld, 2016. Т. 6, № 144. С.69-72.
2. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Модели информационного управления в социальных сетях // Проблемы управления. 2009. № 5. С.28-35.
3. Granovetter M. Threshold Models of Collective Behavior // American Journal of Sociology, 1978. Vol. 83, No 6. P. 1420-1443
4. Rolfe M. Social Networks and Threshold Models of Collective Behavior. Preprint. – Chicago: University of Chicago, 2004
5. Kempe D., Kleinberg J., Tardos E. Maximizing the Spread of Influence through a Social Network / Proceedings of the 9-th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003. P. 137-146.
6. Morris S. Contagion // The Review of Economic Studies, 2000. Vol. 67, No 1. P. 57-78.
7. Valente T. Network Models of the Diffusion of Innovations, Cresskill, NJ: Hampton Press, 1995.
8. Romualdo P., Alessandro V. Epidemic Spreading in Scale-Free Networks // Physical Review Letters. 2001. No 14(86). P. 3200-3203.
9. Tarnowe. Like Water and Vapor – Conformity and Independence in the Large Group. <http://cogprints.org/4274/1/LargeGroupOrderTarnow.pdf>
10. Goldenberg J., Libai B., Muller E. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth // Marketing Letters, 2001. No 2. P. 11-34.
11. Zhang D., Gatica-Perez D., Bengio S., ROY D. Learning Influence among Interacting Markov Chains // Neural Information Processing Systems (NIPS), 2005. P. 132-141.

12. Tuomela R. Shared Belief.
<http://www.valt.helsinki.fi/staff/tuomela/papers/Shared.pdf>
13. Bramoulle Y., Kranton R. Public Goods in Networks // Journal of Economic Theory, 2007. Vol. 135(1). P. 478-494
14. Chwe M. S. Communication and Coordination in Social Networks // Review of Economic Studies. 2000. No 67. P. 1-16.
15. Janky B., Takacs K. Social Control, Participation in Collective Action and Network Stability. – HUNNET Working Paper. 2002.
<http://www.socialnetwork.hu/>.
16. Губанов Д. А., Новиков Д. А., Чхартишвили А. Г. Модели информационного влияния и информационного управления в социальных сетях // Проблемы управления. 2009. №5. С. 28-35.
17. Губанов Д. А., Новиков Д. А., Чхартишвили А. Г. Модели репутации и информационного управления в социальных сетях // Математическая теория игр и ее приложения. 2009. Т. 1, Вып. 2. С. 14-37.
18. Domingos P., Richardson M. *Mining the Network Value of Customers* / Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining. 2002. P. 57-66.
19. Even-Dar E., Shapira A. A Note on Maximizing the Spread of Influence in Social Networks / Internet and Network Economics. 2007. P. 281-286.
20. Watts D., Dodds P. Influentials, Networks, and Public Opinion Formation // Journal of Consumer Research. 2007. No 34. P. 441-458.
21. Carnes T., Nagarajan C., Wild S. M., Zuylen A. Maximizing Influence in a Competitive Social Network: A Follower's Perspective / Proceedings of the Ninth International Conference on Electronic Commerce. 2007. P. 351-360.

Приложения

Код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Social_network
{
    class Program
    {
        static string ReadN()
        {
            string str = Console.ReadLine();
            double num;
            bool isNum = double.TryParse(str, out num);
            if (isNum)
            {
                return str;
            }
            else
            {
                Console.WriteLine("Некорректный формат строки. Повторите попытку");
                return ReadN();
            }
        }
        static void Main(string[] args)
        {
            int t;
            Console.WriteLine("Введите количество агентов:");
            int n = Int32.Parse(ReadN());
            Agents Agent = new Agents();
            double[] kf = new double[8];
            double[] Opinion = new double[n];
            double[] Opinion2 = new double[n];
            double[,] Opinion1 = new double[n, n];
            int[, ,] DArr = new int[n, n, 8];
            double[] Lower = new double[n];
            double[] Upper = new double[n];
            double[,] influence = new double[n, n];
            double[,] influence1 = new double[n, n];
            double[,] influence2 = new double[n, n];
            double[] summ = new double[n];
            double summ1 = 0;
            double max;
            double rv0;
            double rv1;
            double[] ELP = new double[n];
            double[] L = new double[n];
            //записываем мнения
            for (int i = 0; i < n; i++)
            {
                Console.WriteLine("Введите мнение агента{0}:", i);
                double opinion = Convert.ToDouble(ReadN());
                Opinion[i] = opinion;
                Opinion2[i] = opinion;
            }
            for (int k = 0; k < n; k++)
            {
                Console.WriteLine(Opinion[k]);
            }
            Console.ReadKey();
        }
    }
}
```

```

//записываем факторы
Console.WriteLine("Введите параметры:");
for (int i = 0; i < n; i++)
{
    for (int k = 0; k < n; k++)
    {
        for (int j = 0; j < 8; j++)
        {
            Console.WriteLine("Введите параметр x_{0}", j, " для агентов {1}
", i, " и {2}", k, ":");
            int Par = Int32.Parse(ReadN());
            DArr[i, k, j] = Par;
        }
    }
}
for (int i = 0; i < 8; i++)
{
    for (int k = 0; k < n; k++)
    {
        for (int j = 0; j < n; j++)
        {
            Console.Write(DArr[j, k, i]);
            Console.Write(" ");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
    Console.WriteLine();
    Console.WriteLine();
}
Console.ReadKey();
//записываем коэффициенты
for (int i = 0; i < 8; i++)
{
    Console.WriteLine("Введите коэффициент kf_", i, ":");
    kf[i] = Convert.ToDouble(ReadN());
}
//записываем начальную матрицу влияний
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        for (int m = 0; m < 8; m++)
        {
            if (i == j) { influence[i, j] = 0; }
            else
            {
                influence[i, j] += DArr[i, j, m] * kf[m];
            }
        }
        influence2[i, j] = influence[i, j];
    }
}
for (int k = 0; k < n; k++)
{
    for (int j = 0; j < n; j++)
    {
        Console.Write(influence[k, j]);
        Console.Write(" ");
    }
    Console.WriteLine();
}
Console.ReadKey();

```

```

//записываем пороги
for (int i = 0; i < n; i++)
{
    Console.Write("Введите нижний порог для агента_{0} ", i, "\n");
    double lower = Convert.ToDouble(ReadN());
    Lower[i] = lower;
    Console.Write("Введите верхний порог для агента_{0} ", i, "\n");
    double upper = Convert.ToDouble(ReadN());
    Upper[i] = upper;
}
for (int k = 0; k < n; k++)
{
    Console.Write(Lower[k]);
    Console.WriteLine(Upper[k]);
}
Console.ReadKey();
//записываем вектор возможностей
for (int i = 0; i < n; i++)
{
    Console.WriteLine("Введите возможность агента_{0}", i, ":");
    Convert.ToDouble(ReadN());
}
Console.ReadKey();
//записываем конечную матрицу влияний
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        if (Opinion[i] >= Upper[i])
        {
            influence[i, j] = influence[i, j] + 1;
        }
        else
        {
            if (Opinion[i] < Upper[i] && Opinion[i] > Lower[i])
            {
                influence[i, j] = 1;
            }
            if (Opinion[i] <= Lower[i])
            {
                influence[i, j] = 1- influence[i, j];
            }
        }
        if (i == j)
        {
            influence[i, j] = 0;
        }
    }
}
for (int k = 0; k < n; k++)
{
    for (int j = 0; j < n; j++)
    {
        Console.Write(influence[k, j]);
        Console.Write(" ");
    }
    Console.WriteLine();
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        influence1[i, j] = influence[i, j];
    }
}

```

```

}
Console.ReadKey();
int o = 1;
while (o == 1)
{
    Console.WriteLine("Введите способ: 1 - с ценностью");
    int h = Int32.Parse(ReadN());
    switch (h)
    {
        case 1:
            //записываем стоимость с учетом/без учета скидки
            Console.WriteLine("Введите стоимость без учета скидки");
            rv0 = Convert.ToDouble(ReadN()); ;
            Console.WriteLine("Введите стоимость с учетом скидки");
            rv1 = Convert.ToDouble(ReadN());
            Console.WriteLine("Введите затраты на маркетинг");
            double c = Convert.ToDouble(ReadN());
            //записываем количество моментов времени
            Console.WriteLine("Введите количество моментов времени");
            t = Int32.Parse(ReadN());
            //вычисляем прибыль без маркетинга
            //моделируем изменение собственных мнений через заданные моменты
времени

            for (int i = 0; i < t; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    for (int k = 0; k < n; k++)
                    {
                        if (j != k)
                        {
                            if (Opinion[j] >= Upper[j])
                            {
                                Opinion1[j, k] = (Opinion[k] * influence[j,
k] - Opinion[k]);
                                Lower[j])
                                Opinion1[j, k] = 0;
                            }
                            if (Opinion[j] < Upper[j] && Opinion[j] >
Lower[j])
                            {
                                Opinion1[j, k] = 0;
                            }
                            if (Opinion[j] <= Lower[j])
                            {
                                Opinion1[j, k] = (Opinion[k] * influence[j,
k] - Opinion[k]);
                            }
                        }
                    }
                }
            }
            for (int j = 0; j < n; j++)
            {
                for (int k = 0; k < n; k++)
                {
                    Opinion[k] += Opinion1[j, k];
                    if (Opinion[k] > 1) { Opinion[k] = 1; }
                    if (Opinion[k] < 0) { Opinion[k] = 0; }
                }
            }
            //изменяем матрицу влияний в соответствии с новыми мнениями
агентов

            for (int k = 0; k < n; k++)
            {
                for (int j = 0; j < n; j++)
                {
                    if (Opinion[k] >= Upper[k])

```

```

        {
            influence[k, j] = influence2[k, j] + 1;
        }
        if (Opinion[k] < Upper[k] && Opinion[k] > Lower[k])
        {
            influence[k, j] = 1;
        }

        if (Opinion[k] <= Lower[k] )
        {
            influence[k, j] = 1-influence2[k, j];
        }
        if (k == j)
        {
            influence[k, j] = 0;
        }
    }
}
//вычисляем выручку
for (int k = 0; k < n; k++)
{
    summ1 += Opinion[k] * rv0 * L[k];
}
//приводим к первоначальным данным
for (int i = 0; i < n; i++)
{
    Opinion[i] = Opinion2[i];
    for (int j = 0; j < n; j++)
    {
        influence[i, j] = influence1[i, j];
    }
}
Console.WriteLine("Введите алгоритм: 1 - одиночный обход;2 - жадный;3 - по выпуклой поверхности");
int h1 = Int32.Parse(ReadN());
switch (h1)
{
    case 1:
        double c1 = 0;
        //считаем ELP для каждого агента
        for (int l = 0; l < n; l++)
        {
            c1 = 0;
            //активируем одного агента
            for (int i = 0; i < n; i++)
            {
                if (l == i)
                {
                    if (Opinion[l] >= Upper[l])
                    {
                        c1 = c;
                        Opinion[l] = 1;
                    }
                    if (Opinion[l] < Upper[l] && Opinion[l] >
Lower[l])
                    {
                        c1 = c;
                        Opinion[l] = Upper[l];
                        for (int j = 0; j < n; j++)
                        {
                            influence[i, j] = influence2[i, j] +
1;
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (Opinion[l] <= Lower[l])
        {
            c1 = c;
            Opinion[l] = Opinion[l] + Upper[l];
            for (int j = 0; j < n; j++)
            {
                influence[i, j] = influence2[i, j]) +
1;
            }
        }
    }
}
//моделируем изменение собственных мнений через
заданные моменты времени
for (int i = 0; i < t; i++)
{
    for (int j = 0; j < n; j++)
    {
        for (int k = 0; k < n; k++)
        {
            if (j != k)
            {
                if (Opinion[j] >= Upper[j])
                {
                    Opinion1[j, k] = (Opinion[k] *
influence[j, k] - Opinion[k]);
                    Opinion[j] > Lower[j])
                {
                    Opinion1[j, k] = 0;
                }
                if (Opinion[j] <= Lower[j])
                {
                    Opinion1[j, k] = (Opinion[k] *
influence[j, k] - Opinion[k]);
                }
            }
        }
    }
}
for (int j = 0; j < n; j++)
{
    for (int k = 0; k < n; k++)
    {
        Opinion[k] += Opinion1[j, k];
        if (Opinion[k] > 1) { Opinion[k] = 1; }
        if (Opinion[k] < 0) { Opinion[k] = 0; }
    }
}
//изменяем матрицу влияний в соответствии с
НОВЫМИ МНЕНИЯМИ АГЕНТОВ
for (int k = 0; k < n; k++)
{
    for (int j = 0; j < n; j++)
    {
        if (Opinion[k] >= Upper[k])
        {
            influence[k, j] = influence2[k, j] +
1;
        }
        if (Opinion[k] < Upper[k] && Opinion[k] >
Lower[k])
        {
            influence[k, j] = 1;
        }
    }
}

```

```

        if (Opinion[k] <= Lower[k] )
        {
            influence[k, j] = 1-influence2[k, j]
        }
        if (k == j)
        {
            influence[k, j] = 0;
        }
    }
}
// считаем ожидаемую прибыль от агента l
for (int k = 0; k < n; k++)
{
    if (k == l)
    {
        ELP[l] += Opinion[k] * rv1*L[k];
    }
    else
    {
        ELP[l] += Opinion[k] * rv0*L[k];
    }
}
ELP[l] = ELP[l] - summ1 - c1;
//приводим к первоначальным данным
for (int i = 0; i < n; i++)
{
    Opinion[i] = Opinion2[i];
    for (int j = 0; j < n; j++)
    {
        influence[i, j] = influence1[i, j];
    }
}
Console.WriteLine(ELP[l]);
Console.ReadKey();
}
double ELP1 = 0;
double minus = 0;
c1 = 0;
//активируем нужных нам агентов
for (int l = 0; l < n; l++)
{
    if (ELP[l] > 0)
    {
        Console.WriteLine("Agent_{0}", l);
        for (int i = 0; i < n; i++)
        {
            if (l == i)
            {
                if (Opinion[l] >= Upper[l])
                {
                    c1 = c;
                    Opinion[l] = 1;
                }
                if (Opinion[l] < Upper[l] && Opinion[l] >
Lower[l])
                {
                    c1 = c;
                    Opinion[l] = Upper[l];
                    for (int j = 0; j < n; j++)
                    {
                        influence[i, j] = influence2[i,
j] + 1;

```



```

    }
    }
    if (Opinion[1] <= Lower[1])
    {
        c1 = c;
        Opinion[1] = Opinion[1] + Upper[1];
        for (int j = 0; j < n; j++)
        {
            influence[i, j] = influence2[i,
j] + 1;
        }
    }
    }
    }
    }
    minus += c1;
    Console.WriteLine("minus:_{0}", minus);
}
}
//моделируем изменение собственных мнений через заданные
моменты времени
for (int i = 0; i < t; i++)
{
    for (int j = 0; j < n; j++)
    {
        for (int k = 0; k < n; k++)
        {
            if (j != k)
            {
                if (Opinion[j] >= Upper[j])
                {
                    Opinion1[j, k] = (Opinion[k] *
influence[j, k] - Opinion[k]);
                }
                if (Opinion[j] < Upper[j] && Opinion[j] >
Lower[j])
                {
                    Opinion1[j, k] = 0;
                }
                if (Opinion[j] <= Lower[j])
                {
                    Opinion1[j, k] = (Opinion[k] *
influence[j, k] - Opinion[k]);
                }
            }
        }
    }
}
for (int j = 0; j < n; j++)
{
    for (int k = 0; k < n; k++)
    {
        Opinion[k] += Opinion1[j, k];
        if (Opinion[k] > 1) { Opinion[k] = 1; }
        if (Opinion[k] < 0) { Opinion[k] = 0; }
    }
}
//изменяем матрицу влияний в соответствии с новыми
мнениями агентов
for (int k = 0; k < n; k++)
{
    for (int j = 0; j < n; j++)
    {
        if (Opinion[k] >= Upper[k] )
        {
            influence[k, j] = influence2[k, j] + 1;
        }
    }
}

```

```

Lower[k])
    if (Opinion[k] < Upper[k] && Opinion[k] >
    {
        influence[k, j] = 1;
    }

    if (Opinion[k] <= Lower[k] && influence[k, j]
    {
        influence[k, j] = 1-influence2[k, j];
    }
    if (k == j)
    {
        influence[k, j] = 0;
    }
    }
}
//считаем выручку при активации агентов
for (int k = 0; k < n; k++)
{
    if (ELP[k] > 0)
    {
        ELP1 += Opinion[k] * rv1 * L[k];
    }
    else
    {
        ELP1 += Opinion[k] * rv0 * L[k];
    }
    Console.WriteLine("ELP:_{0}", ELP1);
}
//вычисляем ожидаемую прибыль
ELP1 = ELP1 -summ1- minus;
Console.WriteLine("Profit:_{0}",ELP1);
//приводим к начальным данным
for (int i = 0; i < n; i++)
{
    ELP1 = 0;
    Opinion[i] = Opinion2[i];
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        influence[i, j] = influence1[i, j];
    }
}
summ1 = 0;
Console.ReadKey();
break;
case 2:
    double s = summ1;
    double[] Opinion3 = new double[n];
    double[,] influence3 = new double[n,n];
    double[] Opinion4 = new double[n];
    double[,] influence4 = new double[n,n];
    double ELP2 = 0;
    double c2 = 0;
    c1 = 0;
    //активируем нужных агентов
    for (int l = 0; l < n; l++)
    {
        ELP2 = 0;
        //запоминаем мнения и влияния агентов до активации

```

одного

```

for (int i = 0; i < n; i++)
{
    Opinion4[i] = Opinion[i];
    for (int j = 0; j < n; j++)
    {
        influence4[i, j] = influence[i, j];
    }
}
//активируем одного агента
for (int i = 0; i < n; i++)
{
    if (l == i)
    {
        if (Opinion[l] >= Upper[l])
        {
            c1 = c;
            Opinion[l] = 1;
        }
        if (Opinion[l] < Upper[l] && Opinion[l] >
Lower[l])
        {
            c1 = c;
            Opinion[l] = Upper[l];
            for (int j = 0; j < n; j++)
            {
                influence[i, j] = influence2[i, j] +
1;
            }
        }
        if (Opinion[l] <= Lower[l])
        {
            c1 = c;
            Opinion[l] = Opinion[l] + Upper[l];
            for (int j = 0; j < n; j++)
            {
                influence[i, j] = influence2[i, j] +
1;
            }
        }
    }
}
c2 += c1;
//запоминаем мнения и влияния агентов при активации
данного агента
for (int i = 0; i < n; i++)
{
    Opinion3[i] = Opinion[i];
    for (int j = 0; j < n; j++)
    {
        influence3[i, j] = influence[i, j];
    }
}
//моделируем изменение собственных мнений через
заданные моменты времени
for (int i = 0; i < t; i++)
{
    for (int j = 0; j < n; j++)
    {
        for (int k = 0; k < n; k++)
        {
            if (j != k)
            {
                if (Opinion[j] >= Upper[j])
                {

```

```

        Opinion1[j, k] = (Opinion[k] *
    }
    if (Opinion[j] < Upper[j] &&
    {
        Opinion1[j, k] = 0;
    }
    if (Opinion[j] <= Lower[j])
    {
        Opinion1[j, k] = (Opinion[k] *
    }
    }
}
for (int j = 0; j < n; j++)
{
    for (int k = 0; k < n; k++)
    {
        Opinion[k] += Opinion1[j, k];
        if (Opinion[k] > 1) { Opinion[k] = 1; }
        if (Opinion[k] < 0) { Opinion[k] = 0; }
    }
}
//изменяем матрицу влияний в соответствии с
НОВЫМИ МНЕНИЯМИ АГЕНТОВ
for (int k = 0; k < n; k++)
{
    for (int j = 0; j < n; j++)
    {
        if (Opinion[k] >= Upper[k] )
        {
            influence[k, j] = influence2[k, j] +
1;
        }
        if (Opinion[k] < Upper[k] && Opinion[k] >
        {
            influence[k, j] = 1;
        }
        if (Opinion[k] <= Lower[k] )
        {
            influence[k, j] = 1-influence2[k, j]
        }
        if (k == j)
        {
            influence[k, j] = 0;
        }
    }
}
}
// считаем ожидаемую прибыль от агента 1
for (int k = 0; k < n; k++)
{
    if (k == 1)
    {
        ELP2 += Opinion[k] * rv1 * L[k];
    }
    else
    {
        ELP2 += Opinion[k] * rv0 * L[k];
    }
}

```

```

    }
    Console.WriteLine("минусkkkkk_{0}", c2);
    ELP2 = ELP2 - summ1 - c2;
    if (ELP2 > s)
    {
        //приводим к первоначальным данным с активацией
        for (int i = 0; i < n; i++)
        {
            Opinion[i] = Opinion3[i];
            for (int j = 0; j < n; j++)
            {
                influence[i, j] = influence3[i, j];
            }
        }
        s = ELP2;
        ELP2 = 0;
        Console.WriteLine("Agent_{0}", 1);
        Console.WriteLine("Profit_{0}", ELP2);
    }
    else
    {
        //приводим к первоначальным данным без активации
        for (int i = 0; i < n; i++)
        {
            Opinion[i] = Opinion4[i];
            for (int j = 0; j < n; j++)
            {
                influence[i, j] = influence4[i, j];
            }
        }
        c2 -= c1;
        ELP2 = 0;
    }
}
// Console.WriteLine("Profit_{0}", ELP2);
//приводим к начальным данным
for (int i = 0; i < n; i++)
{
    Opinion[i] = Opinion2[i];
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        influence[i, j] = influence1[i, j];
    }
}
summ1 = 0;
Console.ReadKey();
break;
case 3:
    int [] g1 = new int[n];
    g1[0] = -1;
    double[] Op1 = new double[n];
    double[,] inf = new double[n,n];
    double[] ELP3 = new double[n];
    Console.WriteLine("Введите нужное количество агентов:");
    int g = Int32.Parse(ReadN());
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            inf[i, j] = influence[i, j];
        }
    }
}

```

```

while (g != 0)
{
    for (int l = 0; l < n; l++)
    {
        c1 = 0;
        //активируем одного агента
        for (int i = 0; i < n; i++)
        {
            if (l == i && l!=g1[i])
            {
                if (Opinion[l] >= Upper[l])
                {
                    c1 = c;
                    Opinion[l] = 1;
                }
                if (Opinion[l] < Upper[l] && Opinion[l] >
Lower[l])
                {
                    c1 = c;
                    Opinion[l] = Upper[l];
                    for (int j = 0; j < n; j++)
                    {
                        influence[i, j] = influence2[i,
j] + 1;
                    }
                }
                if (Opinion[l] <= Lower[l])
                {
                    c1 = c;
                    Opinion[l] = Opinion[l] + Upper[l];
                    for (int j = 0; j < n; j++)
                    {
                        influence[i, j] = influence2[i,
j] + 1;
                    }
                }
            }
        }
    }
    //моделируем изменение собственных мнений через
заданные моменты времени
    for (int i = 0; i < t; i++)
    {
        for (int j = 0; j < n; j++)
        {
            for (int k = 0; k < n; k++)
            {
                if (j != k)
                {
                    if (Opinion[j] >= Upper[j])
                    {
                        Opinion1[j, k] = (Opinion[k]
* influence[j, k] - Opinion[k]);
                    }
                    if (Opinion[j] < Upper[j] &&
Opinion[j] > Lower[j])
                    {
                        Opinion1[j, k] = 0;
                    }
                    if (Opinion[j] <= Lower[j])
                    {
                        Opinion1[j, k] = (Opinion[k]
* influence[j, k] - Opinion[k]);
                    }
                }
            }
        }
    }
}

```

```

    }
    for (int j = 0; j < n; j++)
    {
        for (int k = 0; k < n; k++)
        {
            Opinion[k] += Opinion1[j, k];
            if (Opinion[k] > 1) { Opinion[k] = 1;
            if (Opinion[k] < 0) { Opinion[k] = 0;

        }
    }
    //изменяем матрицу влияний в соответствии с
    новыми мнениями агентов
    for (int k = 0; k < n; k++)
    {
        for (int j = 0; j < n; j++)
        {
            if (Opinion[k] >= Upper[k])
            {
                influence[k, j] = influence2[k,

            }
            if (Opinion[k] < Upper[k] &&

            {
                influence[k, j] = 1;
            }

            if (Opinion[k] <= Lower[k] )
            {
                influence[k, j] = 1-influence2[k,

            }
            if (k == j)
            {
                influence[k, j] = 0;
            }
        }
    }
}
// считаем ожидаемую прибыль от агента l
for (int k = 0; k < n; k++)
{
    if (k == l)
    {
        ELP3[l] += Opinion[k] * rv1 * L[k];
    }
    else
    {
        ELP3[l] += Opinion[k] * rv0 * L[k];
    }
}
ELP3[l] = ELP3[l] - summ1 - c1;

//приводим к первоначальным данным
for (int i = 0; i < n; i++)
{
    if (i != g1[i])
    {
        Opinion[i] = Opinion2[i];
        for (int j = 0; j < n; j++)
        {
            influence[i, j] = influence1[i, j];
        }
    }
}
j] + 1;

Opinion[k] > Lower[k])

j]];

```

```

    }
    else
    {
        Opinion[i] = Op1[i];
        for (int j = 0; j < n; j++)
        {
            influence[i, j] = inf[i, j];
        }
    }
}
if (l == g1[l])
{
    ELP3[l] = 0;
}
}
int d=1;
max = ELP3.Max();
for (int l = 0; l < n; l++)
{
    if (ELP3[l] == max && d==1)
    {
        d=2;
        Console.WriteLine("Agent_{0}", l);
        Console.WriteLine("Profit_{0}", ELP3[l]);
        ELP3[l] = 0; g--;
        //активируем нужного нам агента
        for (int i = 0; i < n; i++)
        {
            if (l == i)
            {
                if (Opinion[l] >= Upper[l])
                {
                    c1 = c;
                    Opinion[l] = 1;
                    for (int j = 0; j < n; j++)
                    {
                        inf[i, j] = influence[i, j];
                    }
                }
                if (Opinion[l] < Upper[l] &&
                    Opinion[l] > Lower[l])
                {
                    c1 = c;
                    Opinion[l] = Upper[l];
                    for (int j = 0; j < n; j++)
                    {
                        influence[i, j] =
                            inf[i, j] = influence[i, j];
                    }
                }
                if (Opinion[l] <= Lower[l])
                {
                    c1 = c;
                    Opinion[l] = Opinion[l] +
                        for (int j = 0; j < n; j++)
                        {
                            influence[i, j] =
                                inf[i, j] = influence[i, j];
                        }
                }
            }
        }
    }
}

```

Opinion[l] > Lower[l])

influence2[i, j] + 1;

Upper[l];

influence2[i, j] + 1;

